openSUSE Kubic

The Container Platform by openSUSE

https://kubic.opensuse.org

Sarah Julia Kriesch

sarah.kriesch@opensuse.org

Agenda

- About me
- The openSUSE project
- openSUSE Kubic
- Features
- Transactional Updates
- openSUSE Leap and Kubic
- Podman
- How to contribute

About me

- Student in Computer Science (B.Sc.), TH Nürnberg
- Student Representative in Senate/ University Council
- Student Representative in AG Study Plan, AG Laboratories
- Founder AG Open Source (Educator + Trainer), TH Nürnberg
- Global Translation Coordinator, QA, Wiki, Advocate (openSUSE) at openSUSE
- Chairman at One Week Experience e.V. (One Week Student)

open build service

openSUSE medical

OSEM

openQA

openSUSE®

Kubic

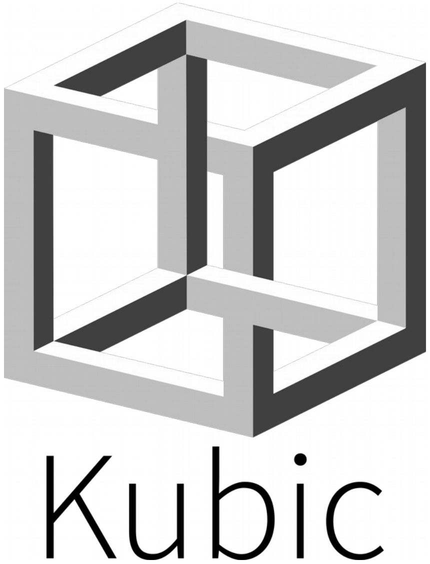
Tumbleweed

FACTORY TESTED

Leap

openSUSE Kubic project

- Container as a Service Platform
- MicroOS, based on openSUSE Tumbleweed
- Provides all to run a Kubernetes cluster



Features

- Transactional Updates
- Fully automated updates possible
- Latest packages with openSUSE Tumbleweed
- Based on Podman
- Easy to use for people with other container knowledge (e.g Docker)

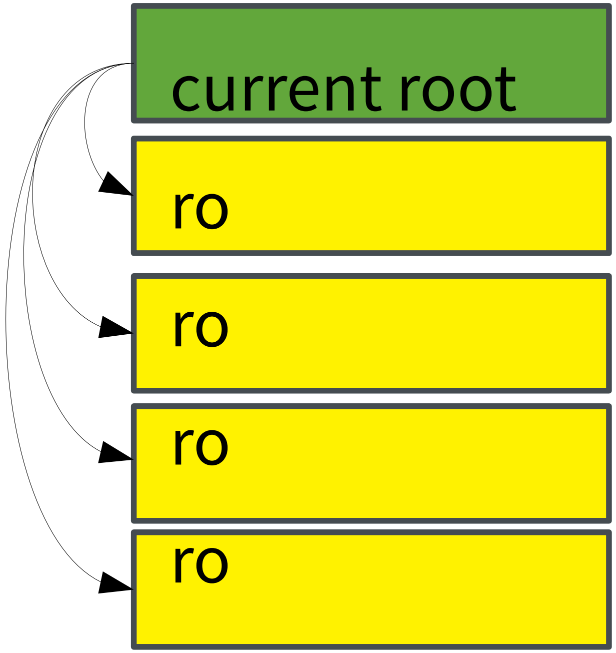
Transactional Updates

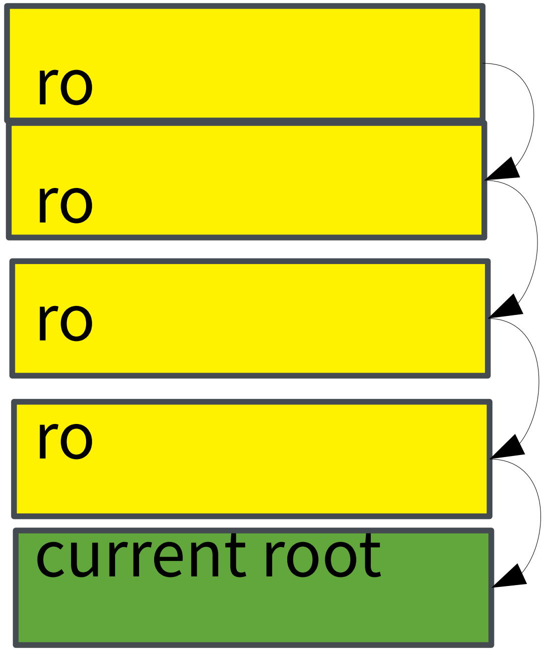
- Is Atomic:
  - Fully applied or not at all
  - Update does not influence the running system
- Can be rolled back:
  - A failed or incompatible update can be discarded and the previous system can be restored

Requirements

- BtrFS root filesystem with snapper (snapshots)/ rollback enabled
- zypper
- btrfsprogs

Snapshots vs. Transactional Updates





Transactional Update

- Clone of current root to ro
- Change rw
- zypper up

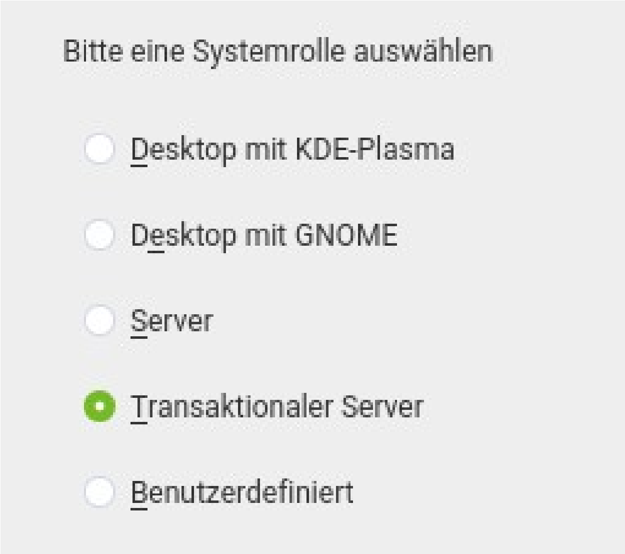
- Change to ro
- “Rollforward” with btrfs subvol set-default
- After reboot change / update executed
- Quickly rollback

Executing Transactional Updates

- Update/Upgrade: transactional-update [up|dup|patch]
- Installation/ Remove / Update 1 package:

  transactional-update pkg [install|remove|update] pkg
- Rollback:

  transactional-update rollback



Bitte eine Systemrolle auswählen

○ Desktop mit KDE-Plasma

○ Desktop mit GNOME

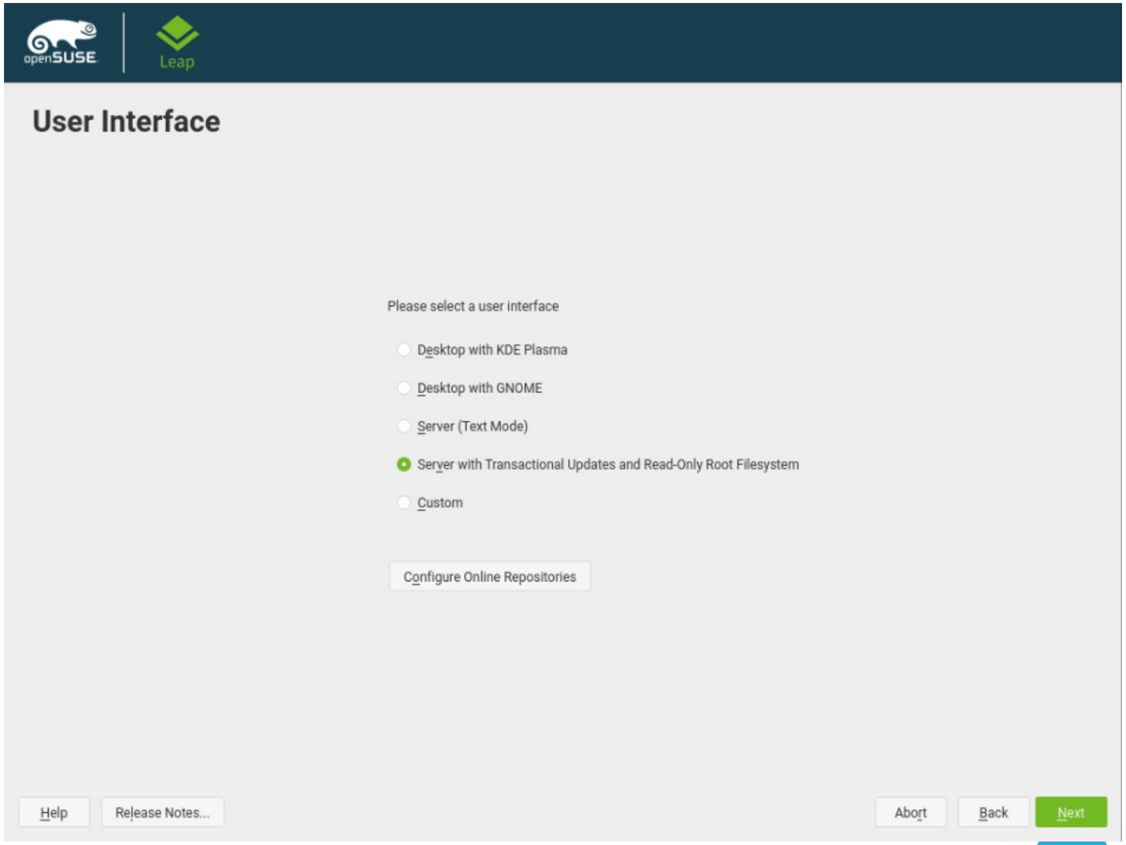
○ Server

● Transaktionaler Server

○ Benutzerdefiniert

How does Leap profit from Kubic?

- Transactional Updates available in Tumbleweed and Leap



User Interface

Please select a user interface

○ Desktop with KDE Plasma

○ Desktop with GNOME

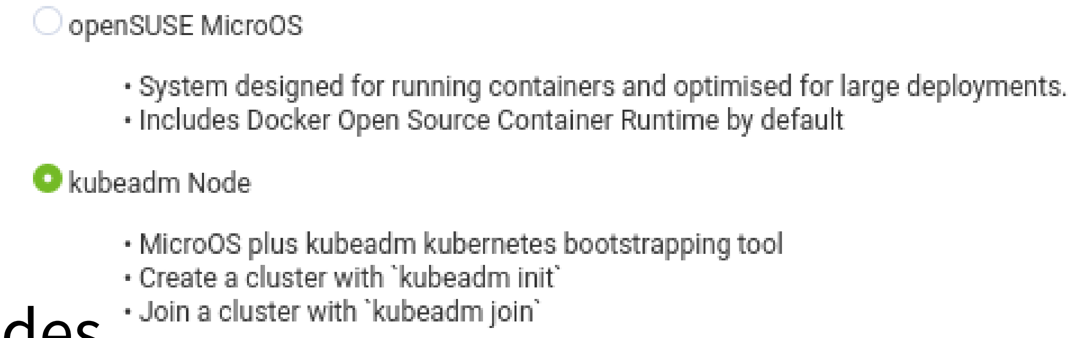
○ Server (Text Mode)

● Server with Transactional Updates and Read-Only Root Filesystem

○ Custom

kubeadm

- toolkit produced by Kubernetes
- Creation of kubeadm nodes for a cluster
- Initialization of the network and joining a node:
- *kubeadm init –pod-network-cidr=10.244.0.0/16*
- *kubeadm join*



○ openSUSE MicroOS

- System designed for running containers and optimised for large deployments.
- Includes Docker Open Source Container Runtime by default

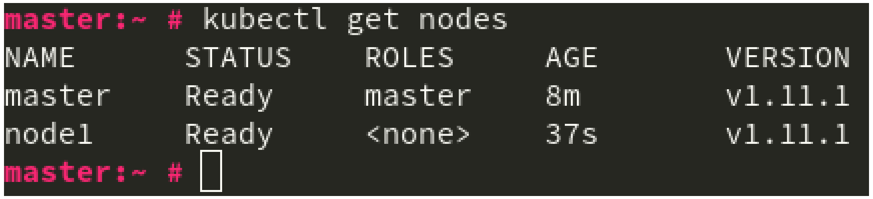
● kubeadm Node

- MicroOS plus kubeadm kubernetes bootstrapping tool
- Create a cluster with `kubeadm init`
- Join a cluster with `kubeadm join`

kubicctl

- Adding Worker Nodes:
  - *kubicctl node add node1,node2,…*
- Verifying the cluster:
  - *kubectl get nodes*

Verification of the cluster



```
master:~ # kubectl get nodes
NAME      STATUS    ROLES     AGE       VERSION
master    Ready     master    8m        v1.11.1
node1     Ready     <none>    37s       v1.11.1
master:~ #
```

Kubic Node Roles

- Kubic Admin Node:
  - kubicd

    (daemon which communicates via gRPC with clients and setup of Kubernetes network)
  - salt-master (for node management via Dashboard)
  - Kubernetes Master Node

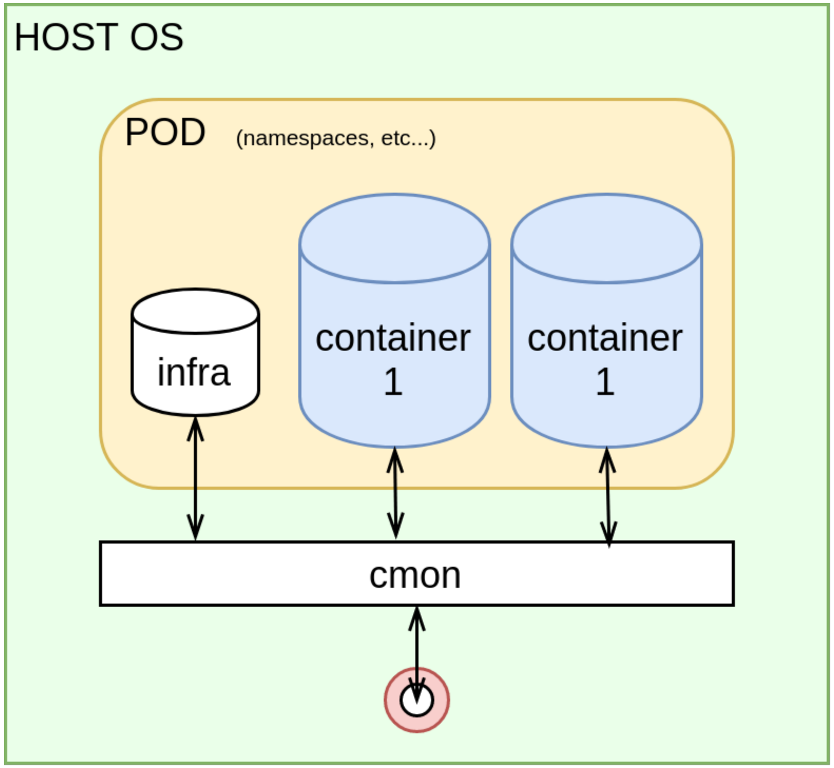
- Kubic Loadbalancer Node:
  - MicroOS without container runtime
  - haproxy installed
- Additional Kubic Node:
  - Additional Master for HA
  - Worker Node

Podman

- Container runtime for running OCI containers
- Same commands as docker
- Daemonless and rootless containers possible

- Container Manager as docker-compose
- Sharing of one network namespace
- Compatible with multiple image formats including the OCI and Docker image formats

Pods



- 1 Kubernetes pod in 1 Linux namespace and the same cgroup
- Flexible combinations of services possible (webserver, github pull, database, ...)

Daemonless

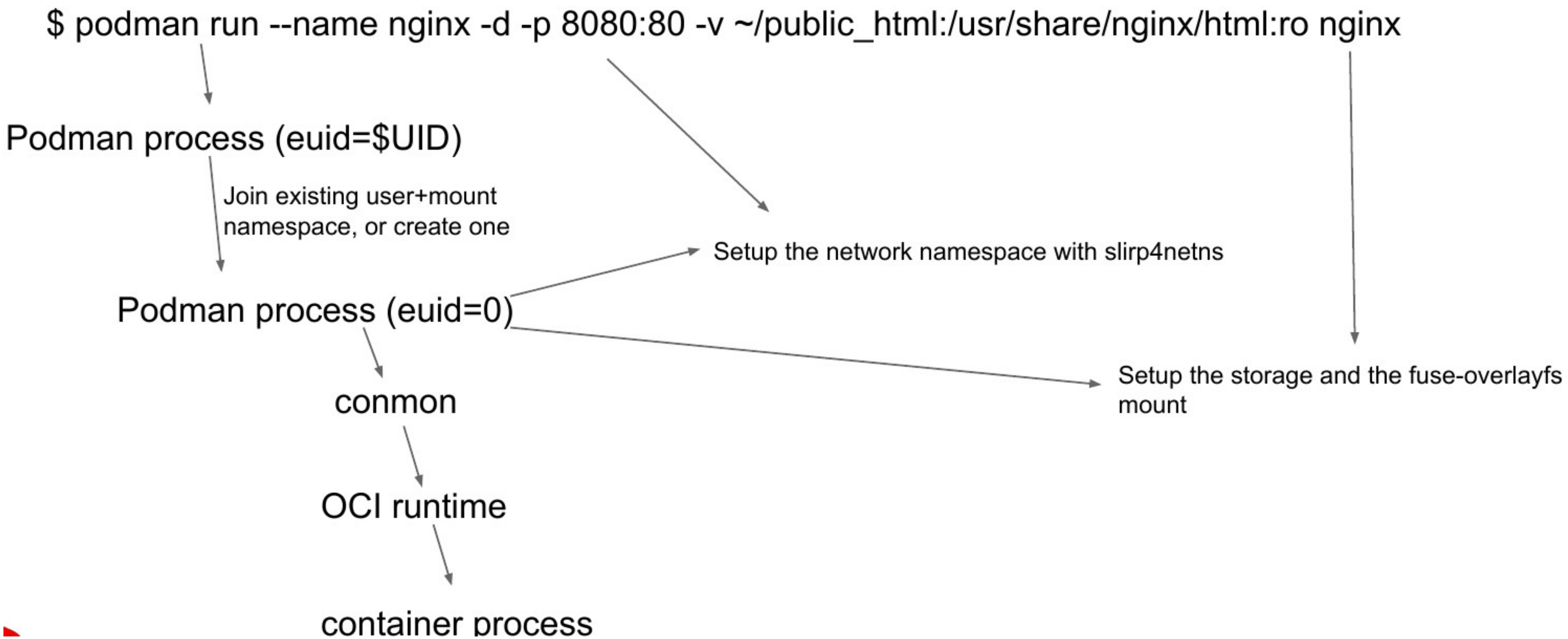
- Docker daemon is (mostly) executed by root
- Podman is working without central daemon
- Additional process *conmon* is monitoring and gives correct start parameters for the container runtime

Rootless

- Start of a container without root privileges
- Mapping of the UID in the namespace
- Using root in a container possible (isolating root)
- User in the container has default access on host system

- Docker offers that since version 19.03

Example of a rootless podman process



```
$ podman run --name nginx -d -p 8080:80 -v ~/public_html:/usr/share/nginx/html:ro nginx
```

Podman process (euid=$UID)

Join existing user+mount namespace, or create one

Podman process (euid=0)

Setup the network namespace with slirp4netns

Setup the storage and the fuse-overlayfs mount

conmon

OCI runtime

container process

Podman commands – as docker

- podman run
- podman exec
- podman info
- podman logs
- podman ps
- podman cp
- podman rm
- podman poll
- podman push
- podman build
- podman image
- podman images
- podman commit
- podman history
- podman stop
- podman start
- podman volume

Podman (only) commands

- podman generate kube
- podman container cleanup
- podman container exists
- podman container restore
- podman image exists
- podman pod create
- podman pod kill
- podman pod ps
- podman pod pause
- podman pod restart

“Do you wonna become a Contributor?”

Packaging

- Creation of packages with the Open Build Service
- Submit to openSUSE Factory
  - > openSUSE Tumbleweed

- https://build.opensuse.org/project/show/devel:kubic

Development of openSUSE Kubic

- Github contributions:
  - https://github.com/kubic-project

Quality Assurance

- Test it!
- Try it!
- Perform it!
- Report bugs!
- Bugzilla: https://bugzilla.opensuse.org/buglist.cgi?component=Kubic&product=openSUSE%20Tumbleweed

References

- openSUSE Kubic:

  https://kubic.opensuse.org
- https://www.projectatomic.io

- Podman:

  https://jaxenter.de/docker/podman-container-daemonless-rootless-89349
- Podman Usage:

  https://github.com/containers/libpod/blob/master/transfer.md

References

- openSUSE Kubic:

  https://kubic.opensuse.org

- Podman:

  https://jaxenter.de/docker/podman-container-daemonless-rootless-89349

- Podman Usage:

  https://github.com/containers/libpod/blob/master/transfer.md

Questions?

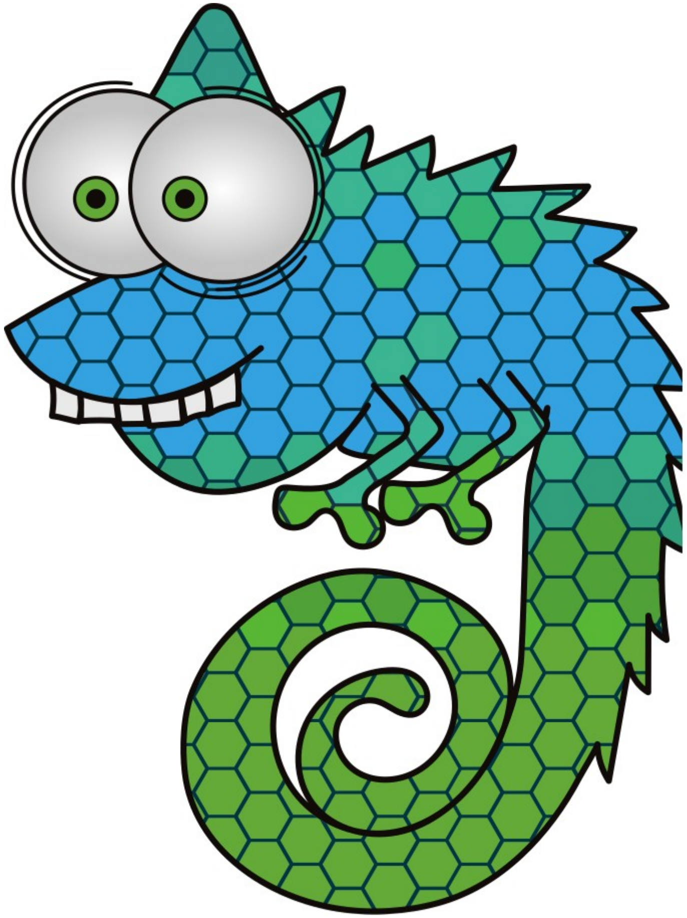
openSUSE Summit 2020
Dublin
May 27 – 28, 2020

https://events.opensuse.org

Join our conference, contribute & have a lot of fun!

events.opensuse.org

Thank you.



Have a Lot of Fun, and Join Us At:

www.opensuse.org

License

General Disclaimer

This document is not to be construed as a promise by any participating organisation to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. openSUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for openSUSE products remains at the sole discretion of openSUSE. Further, openSUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All openSUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC, in the United States and other countries. All third-party trademarks are the property of their respective owners.

Credits

Template
Richard Brown
rbrown@opensuse.org

Design & Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/